

Tesina de Grado  
para la obtención del título de  
Licenciado en Ciencias de la Computación

# Seguridad en iOS y Android: un Análisis Comparativo

---

Autor

**Raúl Ignacio Galuppo**

raul.i.galuppo@gmail.com

G-3483/5

Director

*Dr. Carlos Luna*



Departamento de Ciencias de la Computación  
Facultad de Ciencias Exactas, Ingeniería y Agrimensura  
Universidad Nacional de Rosario  
Diciembre de 2017

# Índice general

<b>1. Hacia un Framework Comparativo</b>	<b>2</b>
1.1. Vista principal . . . . .	2
1.1.1. Funciones no compatibles . . . . .	4
1.2. Catálogo de test . . . . .	4
1.2.1. Contactos . . . . .	4
1.2.2. Calendario . . . . .	5
1.2.3. Geolocalización . . . . .	6
1.2.4. SMS . . . . .	7
1.2.5. Sensores . . . . .	8
1.2.6. Almacenamiento . . . . .	9
1.2.7. DeviceInfo . . . . .	10
1.2.8. Internet . . . . .	11

# List of Algorithms

1.	Test de Contactos. . . . .	5
2.	Test del Calendario. . . . .	5
3.	Test de Geolocalización. . . . .	7
4.	Test de SMS. . . . .	7
5.	Test de los Sensores. . . . .	8
6.	Test de Almacenamiento. . . . .	9
7.	Test de Informacion del Dispositivo. . . . .	10
8.	Test de conexión a Internet. . . . .	11

# Capítulo 1

## Hacia un Framework Comparativo

Android e iOS permiten cambiar ciertos permisos de una aplicación en tiempo de ejecución, es decir, luego de haberla instalado en el dispositivo. Para poner a prueba los sistemas de permisos de ambas plataformas, se ha desarrollado un framework que permite comparar formalmente diversos aspectos de seguridad.

En las siguientes secciones se detallaran los distintos tests que componen el framework. Además, se mencionarán las conclusiones arribadas luego de correr los tests mencionados anteriormente.

### 1.1. Vista principal

El framework es una aplicación móvil y esta compuesto por varios tests. Cada test pone a prueba a un componente del dispositivo, permitiendo así conocer el alcance de los permisos correspondientes a dicho componente. Todos los test fueron implementados en *JavaScript* y corren en el dispositivo mediante *Apache Cordova*<sup>1</sup>.

Al iniciar la aplicación, lo primero que se observa son observan dos áreas principales: Acciones y Test.

La primera área contiene un botón para acceder a la configuración de los permisos del dispositivo. Allí, el *tester* puede cambiar manualmente los permisos requeridos por la aplicación. Además, se encuentra un botón para limpiar la consola (que se encuentra en el otro área).

Mientras que la segunda área se subdivide en dos: en la parte de los tests y la parte de la consola. Una parte corresponde a los botones de los tests que, al presionarse, ejecutan el respectivo test, mostrando en la consola el resultado. Dicho resultado se mostrara con tipografía color verde si fue

---

<sup>1</sup>TODO: va la referencia hacia donde se explica que es Apache Cordova.

exitoso; en cambio, se mostrara con tipografía color roja de ser fallido.  
En la Figura 1.1 se observan las áreas del framework.

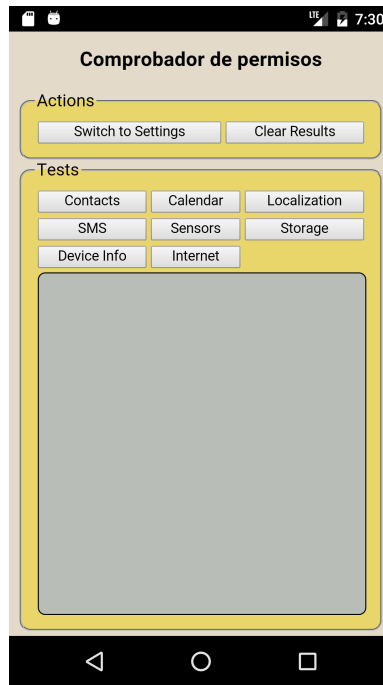


Figura 1.1: Áreas de la aplicación *Runtime Permissions Test*.

A continuación se mencionan los componentes que se pueden testear con el framework:

- Contactos
- Calendario
- Geolocalización
- SMS
- Sensores
- Almacenamiento
- Información del dispositivo.
- Acceso a Internet

### 1.1.1. Funciones no compatibles

El emulador oficial de Android es compatible con la mayoría de las funciones de un dispositivo, pero no incluye la posibilidad de virtualizar los siguientes componentes [1]:

- WiFi;
- Bluetooth;
- NFC<sup>2</sup>;
- Manipulación de la tarjeta SD;
- Conexión USB;
- Micrófono;
- Cámara

Al no poder manipular la tarjeta SD, no es posible testear ninguna de las funcionalidades multimedia: no se pueden grabar audio, ni video ni sacar fotos.

Por lo tanto, no se agregaron al framework tests para los componentes listados anteriormente.

## 1.2. Catálogo de test

En esta sección se listaran todos los test que conforman al framework. Para cada test se detallara su algoritmo, los plugins de Apache Cordova que se utilizaron para confeccionarlo y una serie de capturas.

Para acceder al panel de configuraciones, se utilizo el siguiente plugin: cordova.plugins.diagnostics v3.1.7

### 1.2.1. Contactos

El test consiste en crear un contacto y luego listar todos los contactos presentes en el dispositivo. En caso de ser exitoso, se muestran los contactos. De lo contrario, se muestra un error.

**Plugin:** cordova-plugin-contacts v2.2.1

---

<sup>2</sup>Del ingles *Near Field Communication*. Es una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos.

---

**Algorithm 1** Test de Contactos.

---

- 1: se listan todos los contactos en la consola.
  - 2: se crea un nuevo contacto.
  - 3: se listan todos los contactos en la consola.
- 



Figura 1.2: Testeando la administración de los contactos.

### 1.2.2. Calendario

El test consiste en crear un evento en un determinado rango de fechas y luego listar todos los eventos dentro del rango. En caso de ser exitoso, se muestran los datos del evento. De lo contrario, se muestra un error.

**Plugin:** cordova-plugin-calendar v4.5.5

---

**Algorithm 2** Test del Calendario.

---

- 1: se crea la fecha *startDate*
  - 2: se crea la fecha *endDate*
  - 3: se crea un evento que empieza en la fecha *startDate* y termina en la fecha *endDate*.
  - 4: se listan en la consola los eventos entre las fechas *startDate* y *endDate*.
-

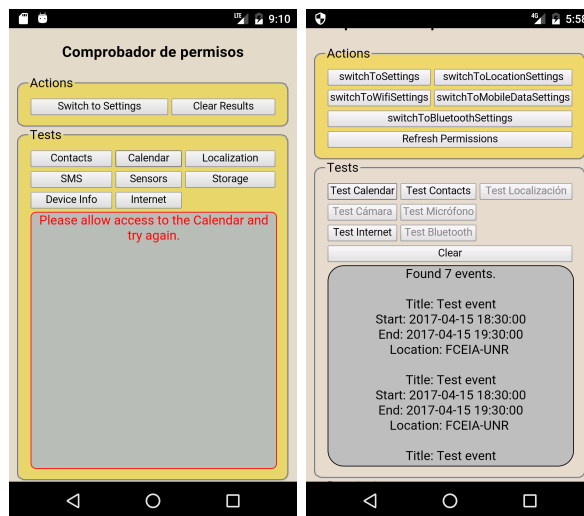


Figura 1.3: Caso exitoso y caso fallido.

### 1.2.3. Geolocalización

Para realizar este test, se configuro el emulador de Android para que simule las coordenadas ( $-122^{\circ}$ ,  $37^{\circ}$ ), tal como se observa en la Figura 1.4. Dicha configuración también se realizo en emulador oficial de iOS.

**Plugin:** cordova-plugin-geolocation v2.4.3

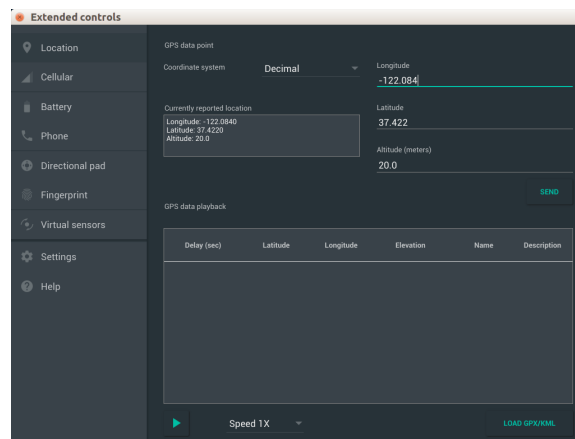


Figura 1.4: Panel de configuración de coordenadas del emulador de Android.



---

**Algorithm 3** Test de Geolocalización.

---

- 1: Se censa el GPS.
  - 2: Se muestran los datos en la consola.
- 



Figura 1.5: Testeando la geolocalización.

#### 1.2.4. SMS

En un principio, se diseñó un test con la capacidad de leer los mensajes. Al momento de implementarlo, se encontró una restricción de seguridad en iOS: a partir de la versión 8 no se pueden acceder a dichos mensajes desde una aplicación instalada por el usuario [2]. En cambio, en Android sí se pueden acceder, siempre que se tengan los permisos correspondientes. Es por ello que se decidió quitar dicha funcionalidad, quedando disponible únicamente la posibilidad de enviar mensajes SMS.

**Plugin:** cordova-plugin-sms v.1

---

**Algorithm 4** Test de SMS.

---

- 1: Se inicializan los eventos para recibir SMS.
  - 2: Se envía un SMS de prueba.
  - 3: Se indica el resultado del test en la consola.
-



Figura 1.6: Test de los permisos de los mensajes SMS.

### 1.2.5. Sensores

El objetivo de este test es obtener datos de dos sensores del dispositivo: del acelerómetro y del giroscopio. Para ello, se configura un `timer`. Durante el tiempo que este activo, se tomaran distintos muestreos; y cuando ocurra el `timeout` se mostraran los datos en la consola de la aplicación.

**Plugin:** cordova-plugin-device-motion

---

#### Algorithm 5 Test de los Sensores.

---

- 1: Se inicializa un `TIMER` con 5 `seg` para detener las mediciones.
  - 2: Se inicia la medición del acelerómetro.
  - 3: Se inicia la medición del giroscopio.
  - 4: Se muestran los resultados en la consola.
- 

**Plugin:** cordova-plugin-gyroscope

En iOS no fueron necesarios permisos para poder correr el test.

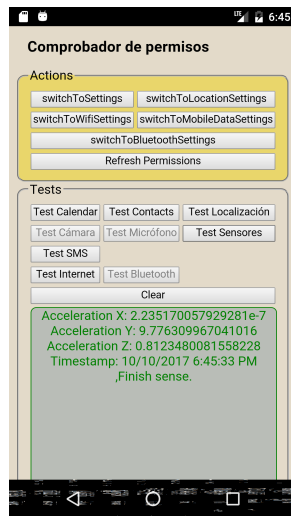


Figura 1.7: Datos medidos.

### 1.2.6. Almacenamiento

El presente test fue diseñado para probar el alcance de los permisos de escritura sobre el sistema de archivos que tiene cada plataforma.

**Plugin:** cordova-screenshot

---

#### **Algorithm 6** Test de Almacenamiento.

---

- 1: Se intenta capturar la pantalla.
  - 2: Se guarda la captura en el dispositivo.
- 

En iOS no fueron necesarios permisos para poder correr el test.



Figura 1.8: Testeando el almacenamiento en el dispositivo.

### 1.2.7. DeviceInfo

El objetivo de este test es obtener datos del dispositivo donde corre la aplicación: plataforma, modelo, serial, entre otros.

**Plugin:** cordova-plugin-device

---

**Algorithm 7** Test de Informacion del Dispositivo.

---

- 1: Se obtienen los datos del dispositivo.
  - 2: **return** un **tag** <IMG>, cuyo **source** es la captura realizada.
- 

No fueron necesarios permisos en ninguna de las dos plataformas para poder correr el test.

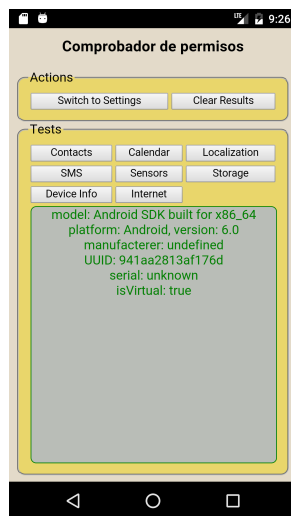


Figura 1.9: Obteniendo datos del dispositivo.

### 1.2.8. Internet

El objetivo del presente test es establecer una comunicación a través de Internet. No se requirió de ningún plugin para implementarlo. La decodificación de la imagen se obtuvo de StackOverflow. Al ejecutarse en un emulador, para probar el acceso a Internet, se habilitó/deshabilitó la Red Inalámbrica de la computadora donde se corrieron los emuladores. No fueron necesarios permisos en ninguna de las dos plataformas para poder

---

#### **Algorithm 8** Test de conexión a Internet.

---

- 1: se realiza una consulta GET HTTP hacia logo del DCC
  - 2: se decodifica la imagen (viene codificada en Base64).
  - 3: **return** un **tag** <IMG>, cuyo **source** es el dato decodificado.
- 

correr el test.

## 1.3. Resultados experimentales



Figura 1.10: Testeando el acceso a Internet.

# Bibliografía

- [1] ANDROID, D. *Developer Android: Funciones no compatibles*. <https://developer.android.com/studio/run/emulator.html#about>.
- [2] APPLE, F. D. *Forum Developer Apple: How to listen for sms reception in ios 8 ?* <https://forums.developer.apple.com/thread/16685>.  
Último acceso 4 de Noviembre del 2017.